Efficient and Location-Private Communication Protocols for WBSNs

AN BRAEKEN[1]*, DAVE SINGELEE[2]

[1] *Department of Industrial Sciences and Technology, Erasmushogeschool Brussel, Belgium*
[2] *ESAT-COSIC, K.U.Leuven & IBBT, Belgium*

Received 16 September 2011; In final form 1 June 2012

To provide healthcare to large heterogeneous populations, one envisions new ways of remote healthcare monitoring in the form of ubiquitous and pervasive healthcare systems. However, the widespread deployment of this technology will depend on the extent to which security and privacy can be guaranteed. In this article, we propose a set of efficient, practical and scalable cryptographic protocols for wireless body sensor networks. Taking into account the computational resources of the sensors, the cryptographic operations performed by the latter exclusively rely on symmetric key cryptography. Our communication protocols guarantee data confidentiality, data authentication, and location privacy of the patient. In addition, we propose an authorization scheme based on time frames. This restricts the doctor's access to the patient's medical data in time, and explicitly enforces the minimal data disclosure principle. Our proposed remote monitoring system outperforms the current state-of-the-art, since it offers security and privacy protection, efficiency and scalability.

Key words: Wireless Body Sensor Network, symmetric key cryptography, data authentication, location privacy

* email: an.braeken@ehb.be

1 INTRODUCTION

The fast and ever increasing development of new wireless technologies and standards is revolutionizing the conventional e-health systems and leads to the migration from fixed to mobile platforms, opening new possibilities and business opportunities. For instance, wireless technologies may ease remote monitoring of life parameters of patients, allow the implementation of teleconsulting services, and drastically improve the spreading and management of clinical information by allowing the creation of a flexible healthcare system, easily deployable in any site.

The development of remote medical monitoring systems will have a large impact on society, as it offers improved healthcare and a significant cost reduction in social security. Considering that the average age of the human population grows significantly, the demand for innovative, high-quality, low-cost medical solution is prominent. In this article we will address the implementation of a wireless body sensor network (WBSN), which offers a remote mobile medical monitoring service. Potential users of this system include people with chronic diseases, homebound and elderly people, etc.

To facilitate widespread adoption of WBSN technology, a number of security and privacy implications must be explored to promote and maintain fundamental medical ethical principles and social expectations [8]. Security vulnerabilities could result in medical harm or even death, since actions by the medical staff will be based on the collected medical data. On the other hand, users will not accept WBSN technology if they could be traced because of the sensors they are carrying, or if their medical data, which is extremely sensitive, is publicly exposed.

Some security schemes have been set up for wireless sensor networks. However, these mechanisms don't work very well in WBSN since there are some subtle differences between these two networks. Firstly, the number of sensors and the range between the different nodes is much more limited in WBSN. Secondly, the sensors are less prone to physical attacks since they are under surveillance of the person carrying them. Last of all, the requirements for privacy are much stronger in WBSN, as discussed before. These specific characteristics, together with the rigid constraints in area, memory, power, energy, etc., should be taken into account.

In the scheme we propose in this article, we have opted for an unbalanced communication protocol, where complex operations are shifted towards the back-end system. The protocol between the sensors and central server relies exclusively on the use of symmetric key cryptography. This makes the so-

lution very suitable to be used in WBSNs. Important design parameters are practicality and scalability, two items which are ignored in several solutions proposed in the literature.

The article is organized as follows. Section 2 describes the setting we envision and the corresponding security aspects and assumptions that we consider for the communication flows between patient, doctor, and central server. These different communication schemes are explained more into detail in Section 3. Section 4 analyzes the security and privacy properties of our scheme. Furthermore, we give a preliminary view on the performance of our solution and demonstrate its feasibility. Section 5 discusses related work and its limitations. Finally, we conclude in Section 6.

2 SYSTEM AND MODEL

We consider the system model in this work. A patient is wearing one or more sensors which form a WBSN. The sensors have a multi-hop connection to the PDA or the mobile phone of the patient [28]. This device is uniquely linked to the patient and is used to manage the set of sensors attached to the patient's body. It is also responsible for forwarding the monitored medical information of the sensors to the central server of the hospital. The forwarding takes place at either irregular times, as a result of a certain trigger, or in case of emergency. When the medical data arrives at the central server, first an analysis is made on the received information. The data is then immediately deleted on the central server and sent to a public database of the hospital, where it is stored in encrypted format. Medical staff from the hospital or a local doctor/nurse are granted access to this data after a successful authorization (e.g. by means of their eID card). To connect to the central server and consult the patient's medical information, they can use a PDA or a mobile phone. The communication setting is illustrated in Figure 1. In the rest of this article, we will discuss how the security and privacy protection of these communication flows can be guaranteed.

2.1 Security and privacy requirements

Since medical data of a patient is sensitive information, it is an interesting target for attacks performed by a malicious third party. One should note that the medical data is available at various places during its entire lifetime. It is sent from the sensors to the server, stored in a public database, and consulted by medical staff. During all these stages, data protection is necessary. This includes [16]:

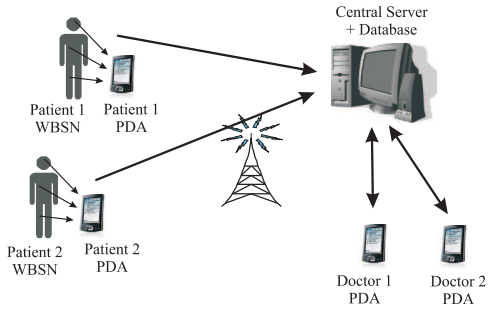

Figure 1
System model

- **Confidentiality:** information should be kept secret from all but those who are authorized to see it.
- **Data authentication:** one should be able to corroborate the source of the information and to be ensured that the information has not been altered by unauthorized or unknown means.

In addition, also location privacy should be ensured. This is often neglected in previous work, but is nevertheless a very important issue. Since the sensors are fixed on the body of the patient, tracking and tracing of the patient becomes relatively straightforward. The use of temporary pseudonyms [9, 17] is a very popular countermeasure to thwart this attack. To avoid additional key sharing and data storage, as is typically required in these schemes and which would increase the cost, we will directly integrate the use of pseudonyms in our security solution.

Note that there are many attacks which can be carried out on the system. An adversary can overhear the wireless communication between sensors, PDA, central server and medical staff. This could potentially lead to interception of the data. Other attacks include altering, deleting or inserting data, replay attacks, denial-of-service attacks, etc. To counter these potential vulnerabilities, it is essential that the medical data is transmitted in an encrypted format, being authenticated (e.g., using a message authentication code), and that freshness of data is guaranteed (using a counter, timestamp or nonce). Not only the wireless communication channel should be protected, but also the storage of the data. Attackers can physically break into one or more components and consequently try to steal the stored information.

2.2 Assumptions

Before we define our communication protocols, it is important to clearly state the assumptions which our solution is based on.

We assume the sensors are tamper resistant. This can be realized in several ways. When the sensor platform consists of both a tamper vulnerable part (e.g., common sensor node hardware) and a tamper resistant part (e.g., a smart card or a cryptographic micro controller), implementation parameters such as price, complexity and power consumption remain reasonable [18]. Another solution for introducing tamper resistance is by exploiting challenge-response pairs of physical uncloneable functions (PUFs) [25, 5] in the communication protocol. These are functions that are embodied in the physical structure of the sensor. The functions are easy to evaluate, i.e., to compute the response for a given challenge. However, they are very hard to characterize due to the randomness of many components in the physical structure and cannot be cloned. Since a PUF depends on the physical molecular structure, any change in that structure will have an effect on the challenge-response pair.

Since the medical data is stored encrypted, it is not necessary to put it in the tamper resistant hardware. The latter only contains the cryptographic keys and other security related material. It should however be stressed that it is quite unlikely that an adversary can access a sensor node attached to the patient's body, without the patient noticing this. As a result, in many practical use cases one can relax the requirements on the tamper resistance and hence reduce the price and complexity.

Also the other devices in the scheme need storage protection. Secret and private keys should be secretly stored. Extra care should be taken on the doctor's PDA, since this device will contain decrypted medical information from the patient. Countermeasures should include secure deletion of data from memory, sandboxing, etc. Since such techniques are commonly used in state-of-the-art implementations, we will not discuss them further in this article. Finally, also the central server should be physically protected since it aggregates, analyzes and stores all the medical information. That is why the data is only stored in encrypted format. This relaxes the requirements on the databases access control enforcement.

Furthermore, we assume that each sensor stores two secret keys: one key shared with the central server, and one key shared with the patient's PDA. Besides these shared keys, both the patient's PDA and the central server have a public/private key pair, which is used to secure their communication. Since these devices are less resource constrained, conventional cryptographic techniques and protocols can be used. We will particularly focus on the commu-

nication protocols with the sensors, as this is a challenging design task.

In the next sections, we will present a practical security solution based on the requirements and assumptions mentioned above. For efficiency reasons, the protection of the communication between sensors and the medical server will be exclusively based on symmetric key cryptography. In addition, we propose a data access control scheme that is not only based on identity, but also on the time of the data retrieval request. This is particularly important in scenarios where the doctor does not need to know the entire medical history of the patient. Such an access control system limits the access to the collected medical data with respect to a certain time, even when the data is stored in a public database. Note that systems with this property of time frame authorization have been previously described in [26], but these rely on the use of less efficient cryptographic techniques.

3 PROPOSED COMMUNICATION SCHEMES

We first discuss some notations and pre-installation steps that need to be performed by the different entities (sensor, patient's PDA, central server, doctor's PDA) in the system. Next, we describe the communication protocol between the sensor and the central server, and between the doctor and the central server, more into detail.

3.1 Notations

For clarity reasons, we will use the following notations in the rest of this article: Se denotes the sensor, CS the central server, P the patient's identity, and D the doctor's identity. The central server stores a secret master key, which is denoted by cs. The private key of the patient's PDA is denoted by pp. S_k represents a signature algorithm using the private key k, E_k an encryption algorithm using the key k, MAC_k a message authentication code under the secret key k, and h a cryptographic hash function. Ctr denotes a counter shared between two or more devices in the network, and sn a sequence number.

3.2 Underlying key agreement protocol

Since the security protocol between the sensors and the central server is based on symmetric key cryptography, an initial secret key has to be shared between the different communication parties. For efficiency reasons, we propose an "identity based" cryptographic approach. The shared secret key is derived from the MAC on the unique hardware ID of the device that is most resource

constrained (e.g., the sensors), while the MAC is computed by the entity that has most resources available (e.g., the central server) using its secret key. For example, the shared key between a sensor and the central server can be represented by $MAC_{cs}(ID_{Se})$. By using this approach, the key can be stored by the former and only needs to be computed on the spot by the latter. Because of this construction, the sensor does not need to perform a complex cryptographic operation such as computing a digital signature. It also improves the scalability, since it avoids the central server storing all secret keys shared with the sensors. One should however note that key update mechanisms need to be employed if a particular key is used for a relatively long period. This is out of the scope of this article.

3.3 Pre-installation

Before communication can take place between sensor and central server, or between the doctor's PDA and the central server, keys need to be pre-installed on the different communication parties. We suppose that this pre-installation is trusted and organized by the hospital. The following steps take place:

- **Sensor:** Each of the sensors securely stores two secret keys: $MAC_{cs}(ID_{Se})$ and $k_{ps} = MAC_{cs}(ID_P \mid\mid ID_{Se})$. The former is shared with the central server, while the latter is shared with the patient's PDA. Without loss of generality, we assume that these keys have the appropriate key sizes. If not, one first has to apply a key expansion or reduction function. Furthermore, each sensor also has a unique counter Ctr, which will serve as a countermeasure against replay attacks.

- **Patient's PDA:** To bind the patient's PDA uniquely to the patient's identity, the device is first registered at the central server (this could even be done remotely, using the patient's electronic identity card). After this step, the patient's identity ID_P and the corresponding public key (e.g., combined in a certificate) are stored on the central server, while the PDA stores the central server's public key. Furthermore, the PDA and the central server share a sequence number sn, used to avoid replay attacks. It could also be replaced by a timestamp. When a sensor is put on the patient's body, the central server sends a message to the patient's PDA that contains the sensors' identity ID_{Se} and the corresponding secret key $k_{ps} = MAC_{cs}(ID_P \mid\mid ID_{Se})$. These keys can be computed on the spot by the central server. For each of the sensors, the PDA also stores the value of the counter Ctr. Note that the data stored on the PDA can be copied to multiple of the user's devices. This

increases reliability and avoids an update mechanism when the PDA is replaced, but also decreases the security level (since the probability of an attacker compromising a device and stealing the keys increases).

- **Central server:** As discussed above, the central server stores a list of the identities of the patients that are registered, together with their public key (or certificate) and the sequence number sn. It also contains a list of the doctors' identities ID_D and an up to date access control list, describing which patient files can be accessed by which medical staff member at which time frame. The determination of this list is out of scope of this article. To maximize scalability, the central server does not need to store a list of sensors put on the patient's body. After this information has been sent to the patient's PDA, the central server deletes its from its memory.

- **Doctor's PDA:** The PDA of the doctor stores a secret key $MAC_{cs}(ID_D)$, which it shares with the central server. These keys are only stored on the doctor's PDA, the central server computes them on the spot. This improves scalability.

Note that this pre-installation is practical and straightforward. Only when a patient enters the hospital and sensors are put on his body, the pre-installation process takes place on the central server and the sensors. The patient manages on his PDA the list of active sensors attached to this body. Each time a sensor is removed from his body, this information is entered in the PDA (e.g., by entering an identifier printed on the sensor during fabrication). Without such an update, an attacker could inject false medical information in the system by using a compromised sensor that is no longer attached to the patient. Note that keeping this list up to date is not a cumbersome process. Sensors are only rarely removed from the patient's body, and the number of sensors in a WBSN is typically rather limited. When a new sensor needs to be added to the WBSN, no changes are required at the other sensors, causing no extra delay.

3.4 Communication between sensor and central server

The first communication channel we will discuss, delivers the monitored medical data from the sensor to the central server, through the patient's PDA. It is a one way communication protocol from sensor to central server. This data is sent at at either irregular times, as a result of a certain trigger, or in case of emergency. Note that a possible trigger can be a request from the

central server to update the latest data or to provide extra info about a certain parameter to improve the treatment. In that case the communication is not purely unilateral.

As mentioned before, each sensor shares a key $k_{ps} = MAC_{cs}(ID_P \mid\mid ID_{Se})$ and a counter Ctr_i with the patient's PDA. The counter is used to ensure data freshness and authorization based on time, and is updated after each communication round. One could also replace the counter by a timestamp, but this is more complex to implement securely and efficiently, particularly on the sensors. When a sensor wants to transmit medical data, when triggered or at irregular time frames i, it first computes a session key K_i corresponding to that particular time frame i. This session key is derived from the shared secret key with the central server $MAC_{cs}(ID_{Se})$.

$$K_i = (k_i \mid\mid k_i') = h(MAC_{cs}(ID_{Se}) \mid\mid Ctr_i).$$

The session key K_i is split in two different parts $(k_i \mid\mid k_i')$. The key k_i will be used for symmetric-key encryption, while the key k_i' will be used for authentication. It is a general, however not proven, rule to take different values for both features.

After having constructed the session key, the sensor sends its monitored data m to the central server. Since the sensor cannot communicate directly to the server, all information in the WBSN is sent to the patient's PDA. More in detail, the following message is sent:

$$E_{k_{ps}}(ID_{Se} \mid\mid Ctr_i), E_{k_i}(m), MAC_{k_i'}(m). \tag{1}$$

After sending the message, the sensor increments the counter by one. When the patient's PDA receives this message, it will decrypt the first part trying all keys k_{ps}. Since the number of sensors in the WBSN is limited, the complexity of this operation is reasonable. Next, it checks the identity of the sensor ID_{Se} and the counter Ctr_i. If the sensor is still part of the WBSN and the counter is strictly larger than the current value stored in the PDA, the message is accepted. Using public-key cryptography, the PDA then constructs the following encrypted message $E_{sc}(ID_P \mid\mid ID_{Se} \mid\mid Ctr_i)$ and the signature $S_{pp}(sn \mid\mid ID_{Se} \mid\mid Ctr_i)$. Here sn denotes the sequence number for communication with the central server. Note that it is important that the signature algorithm does not have a message recovery functionality, to avoid tracking attacks. After having constructed these messages, they are concatenated to the data that was sent by the sensor. More in detail, the following message is

sent to the central server:

$$E_{k_i}(m), MAC_{k'_i}(m), E_{sc}(ID_P \mid\mid ID_{Se} \mid\mid Ctr_i), \\ S_{pp}(sn \mid\mid ID_{Se} \mid\mid Ctr_i). \quad (2)$$

After sending the message, also the patient's PDA augments the counter of the corresponding sensor and the sequence number sn. When receiving this message, the central server first decrypts the message that was encrypted using its public key and obtains the identity of the patient ID_P. Then it checks the validity of the signature by using the public key of the patient, and verifies that the sequence number sn is strictly larger than the value stored in memory. If these checks are successful, it knows that the message was approved by the patient's PDA and is not replayed (because of the sequence number). Next, the sequence number is increased by one, and the shared key $MAC_{cs}(ID_{Se})$ is computed. By combining it with the counter Ctr_i, the central server can compute the session key K_i. Using this session key, the message $E_{k_i}(m)$ is decrypted and the MAC can be verified (to be ensured of the authenticity and the integrity of the data). The result of the decryption is the monitored data m. The server then analyzes this medical data, alarms the doctor when necessary, and puts the data in the public database (see next paragraph). Instead of signing data with its private key, the PDA could also carry out a key agreement protocol with the central server. The outcome of this protocol is a secret session key, which then can be used to guarantee the authenticity the medical data sent to the central server. This solution will not be discussed further in this article.

The communication protocol between the sensor, patient's PDA and the central server is summarized in Fig 2.

3.5 Medical data stored in the database of the central server

After checking the authenticity of the medical data, the central server puts it in a public database. Authorized medical staff can consult this database when necessary (see next paragraph for the protocol details). Since access to the medical data can be restricted in time, it is important to add a timestamp to the data. To link the time to a communication round, the central server computes a timestamp t_i corresponding to the value of the sequence number sn. More in detail, the following information is stored in the database:

$$ID_P, t_i, E_{sc}(ID_{Se} \mid\mid Ctr_i), E_{k_i}(ID_{Se} \mid\mid ID_P \mid\mid t_i \mid\mid m), \\ MAC_{k'_i}(E_{k_i}(ID_{Se} \mid\mid ID_P \mid\mid t_i \mid\mid m))$$

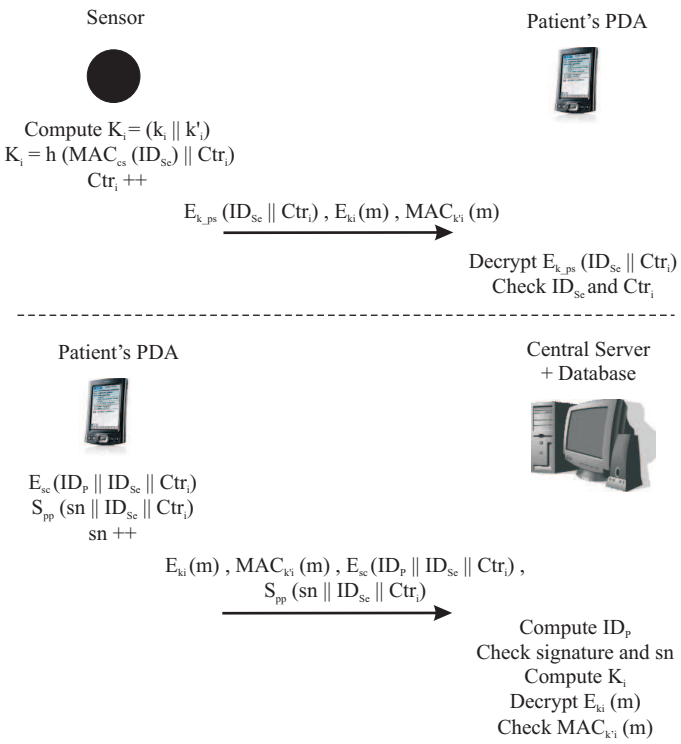

Figure 2
Communication from sensor to central server

Note that ID_P is included in plaintext for efficiency reasons. If the central server needs to send information to the doctor afterwards, the server which contains the database will have to search through the whole database to find the data for the corresponding patient. Now it can just search for the correct patient and time frame, and then decrypt the first encrypted message to compute the session key, which is needed to decrypt and check the authenticity of the rest of the message. This key can then be transported to the doctor's PDA (see next paragraph). In fact, ID_P represents a meaningless number for an outsider. Only the patient and the doctor possess the linking information between the number ID_P and the name of the patient. Even if an attacker is able to link the number ID_P to a patient's name, no further medical information of the patient is leaked since the rest of the database is encrypted.

Of course, this solution is not perfectly privacy prone and is made in a trade-off with efficiency. It is possible to simply remove that information, but as said before, this would require a complete search through the whole database.

3.6 Communication between doctor's PDA and central server

The communication channel between the central server and the PDA of the doctor is bidirectional. In case of an emergency, the central server needs to contact a doctor, taking into account the necessary authorization requirements. The other situation is that a doctor wants to consult the information before or after visiting the patient.

As denoted before, cs is the private key of the central server, and ID_D the unique ID of the doctor. The shared secret key, which is stored in the central server, is equal to $MAC_{cs}(ID_D)$. This secret key is stored in the secure memory of the doctor's PDA, and can be computed on the spot by the central server. The keys used for encryption and authentication respectively are $k_d \mid\mid k'_d = MAC_{cs}(ID_D)$. Let us now discuss the communication protocols (for both directions of the communication flow) more in detail.

Communication initiated by the central server This communication is organized in at least two phases, and in general in three phases, as will be briefly discussed now. The detailed cryptographic operations are presented in Figure 3.

- During the first phase, the central server alarms an authorized doctor by sending an emergency message m_e, using the shared secret key $MAC_{cs}(ID_D)$. As a countermeasure against replay attacks, a timestamp t_i is used instead of a counter since both communication parties possess more computer power and resources (compared to a sensor). By using a timestamp, there is no need to store a synchronized counter.

- The doctor first looks up the secret key and then checks the validity of the timestamp and the MAC. Next, he replies with a message m', concatenated with a timestamp t'_i. There are three possible answers $m' = \{m'_f, m'_n, m'_e\}$. If the verification of the timestamp and/or MAC fails, $m' = m'_f$. If the doctor is unable to react at that moment, he replies with $m' = m'_n$. If he wants to react positively to the request, the reply is $m' = m'_e$.

- If the doctor has replied with m'_e, and the checks on the timestamp and the MAC are valid, the central server computes the session keys

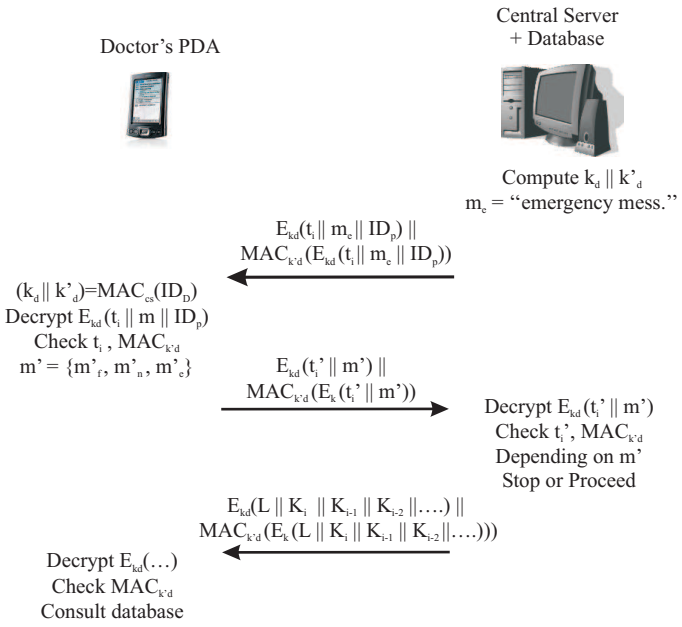

Figure 3
Communication from central server to doctor

needed to consult the necessary medical data stored in the database. These cryptographic keys are sent to the doctor's PDA. Access is given for a fixed number of time slots. The corresponding set of pointers L to the entries in the database are also sent to the doctor, to facilitate the search in the database.

Initiated by the doctor's PDA The secret key shared between the doctor's PDA and the central server is, as already discussed above, $k_d \ || \ k'_d = MAC_{cs}(ID_D)$. We shortly describe the two secure communication steps. The detailed cryptographic operations of these phases are presented in Figure 4.

- During the first phase, the doctor submits a request to the central server, containing a timestamp and the identity of the patient and/or sensor(s). The request is encrypted using the key k_d and its authenticity is protected by computing a MAC (using key k'_d). Without loss of generality,

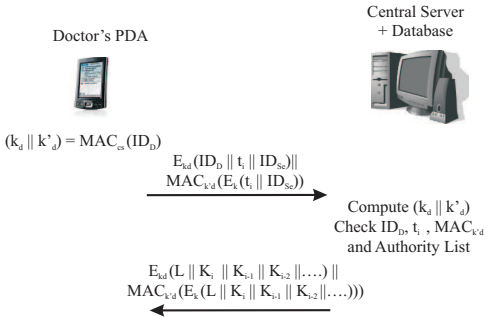

Figure 4
Communication from doctor to central server

we only focus on the situation where the doctor wants to obtain the information from one sensor ID_{Se}. The solution where the doctor requests information from one particular patient is very similar.

- The central server first exhaustively tries all the different keys, corresponding to the list of doctors, until it finds a match. Then it checks the timestamp, MAC, and the authorization privileges of the doctor. When all checks are positive, the cryptographic keys and pointers for consulting the encrypted database are returned (this is identical to the last phase in the previous communication scenario).

4 ANALYSIS

The design of practical, efficient, scalable, secure and privacy-preserving cryptographic protocols for remote monitoring applications will enable the widespread of WBSN technology. In this section, we will discuss some important features of our proposed solution.

4.1 Security discussion

The main security properties of our system are as follows:

- When a sensor node is compromised by an attacker or not needed anymore, it will be removed from the sensor network and also from the list of active sensors, stored and managed at the patient's PDA. Since each

sensor shares a different key with the PDA and central server, cryptographic material from a compromised node cannot be used to carry out attacks on other nodes.

- When cryptographic keys are used for a relatively long period, they should be updated. This avoids some attacks on the cryptographic primitives used in the protocols, which could even result in the compromise of secret or private keys. For instance, if an adversary can compromise the private key of a device, he could decrypt all the exchanged messages, including these from the past. A good guideline is to change the private key regularly, e.g. every year. Since this is standard practice, it will not be discussed further in this article.

  Also the counter and the sequence number should be renewed when their maximum value is reached in order to avoid replay attacks. However, in realistic settings, this will never be the case. Suppose, for instance, a sensor submitting every minute a message. For 32-bit counters starting from 0, it will take more than 8000 years before the counter reaches its maximum value. Having 10 sensors in the network, also a 32-bit sequence number will last more than 800 years.

- Since the medical data is encrypted with a shared key, it remains confidential to eavesdroppers. The secret keys used to secure the communication channel between sensor and PDA or central server cannot be derived by a passive eavesdropper. This follows from the properties of the cryptographic building blocks used in the protocol, and from the fact that a secret key shared between A and B can only be computed by means of a MAC of A on the identity ID_B of B. Since the keys are either computed on the spot, or stored in secure memory, an attacker cannot steal them.

- The integrity and authenticity of the data is guaranteed since every transmitted message contains a MAC on the content of the message. The correctness of the data source can be guaranteed by the fact that the secret session key is only known by the sender and the intended receiver.

- We used timestamps and counters to avoid replay attacks in our communication protocol. A particular attack we prevent in our protocol, is old or fake medical data being injected in the system. A compromised

sensor cannot inject fake data, because the patient's PDA will not accept data from that sensor. Sending it directly to the central server is also not possible, since the compromised sensor cannot spoof the signature of the patient's PDA. An attacker can also not inject old medical data in the system. The patient's PDA will detect this replay attack because the shared counter Ctr_i is not higher than the stored value. Sending the data directly to the central server is not possible. The latter will reject the signature of the PDA embedded in the message, since it does not contain a fresh sequence number.

- Only sensors that were initialized during the pre-installation process, received shared secret keys. Other sensors are not in the list of active sensors, and do not share a key with the patient's PDA. They can hence not communicate with this device. Sending information directly to the central server is not possible, since the MAC of the patient's PDA on the sensor's identity ID_{Se} is required. Due to the construction of the messages in the protocol, all data sent by the PDA (see Eq. 2) is linked to each other.

4.2 Privacy discussion

To achieve location privacy, none of the entities in the system puts its identity in the messages it transmits. This avoids an attacker tracing a patient or a doctor by the data that his device(s) broadcast. Since the identities are not directly put in the message, the receiver has to perform an exhaustive search on the list of possible communication parties. This is however still a reasonable operation. The patient's PDA only communicates to a small number of sensors (the size of a WBSN is typically rather limited), and the number of doctors registered at the central server is also assumed to be rather low. Also note that the exhaustive search is carried out by the communication party with most computing power. This unbalance is necessary to maximize the overall efficiency.

However, since the number of patients registered at a hospital could be relatively large, we used a different mechanism to protect the communication flow between the patient's PDA and the central server. The patient's PDA puts its identity (and the one of the sensor) in an encrypted message. Since this message also contains the counter Ctr_i, which is never used twice, the ciphertext will be random for an outsider and can hence not be used for tracking.

Note that the data from the sensors to the central server is not sent at regular time frames since that would facilitate to trace the patient and thus be in

conflict with the privacy restrictions.

4.3 Scalability

Scalability is one of the main requirements when designing a remote monitoring system. When many patients are registered at the central server, the number of sensors that transmit their medical data could become quite large. Therefore, it is not scalable to manage all the WBSNs from one central location (i.e., the central server). That is why we distributed this functionality into the system. The patient's PDA is responsible for managing its sensor network where it is connected to. Since the number of sensors in the WBSN is limited, this is a reasonable task. The central server then manages the patients' PDAs from which it gets the medical data. The magnitude of the number of patients is much smaller than the number of sensors. Furthermore, the central server also manages the list of doctors which can consult the medical data. Also this list is assumed to be manageable.

Note that our solution can easily be transformed into a solution where the sensors directly send their data to the central server (via a gateway such as a mobile phone). This is possible because the shared key between the sensor and the PDA can also be computed by the central server. The disadvantage of this direct forwarding mechanism, is that the central server should have to manage the list of active sensors. This is less scalable than our original solution. One can also opt for a hybrid scheme, where some of the WBSNs are managed by the PDAs of the patients, and other sensor networks directly by the central server.

4.4 Performance discussion

Since sensors are resource constrained, efficiency is very important. We will now show that our proposed protocols can be realized on current state-of-the-art sensor platforms. To demonstrate the feasibility, we will select some appropriate cryptographic primitives, and give an estimation of the performance in this scenario. Note that we do not intent to give exact numbers on the memory and energy consumption, we only want to show that our solutions can be implemented using off-the-shelf components and conventional cryptographic algorithms.

In our remote monitoring system, the body sensors clearly have the most rigid resource constraints. That is why we will now focus on the cost of introducing cryptographic primitives on these devices. The body sensors are either worn by or implanted in a patient. As a consequence, this requires a very low power consumption to minimize radiation and maximize the lifetime. Moreover, these sensors also have very limited computational power and memory.

	TI Node	MICAz Node	MyriaNed
CPU	16bit, 8MHz	8bit, 16MHz	16bit, 32MHz
RAM	2KB	4KB	8KB
Flash Memory	64KB	128KB	128KB
Voltage	$1.8 \sim 3.6V$	$2.7 \sim 3.3V$	$1.6 \sim 3.6V$
OS	TinyOS	TinyOS	MyriaCore

Table 1
Specifications of some embedded micro controllers [6]

Table 1, published by Gong et al. [6], summarizes the specifications of some embedded micro controllers, often used in sensor networks: TI node, MICAz node, and MyriaNed.

Before discussing the details of the cryptographic algorithms, we first need to fix the security level, taking into account the trade-off between performance and computational resources. Depending on the use case scenario, a security level of 64 bit or 128 bit is acceptable for resource constrained sensors. In this article, we will compare the cost for both these security levels, based on the results of [6] for implementations on the MICAz nodes (TinyOS version 2.10).

The protocols have been designed in such a way that the largest part of the cryptographic computations and storage requirements are shifted towards to the central server. There are only two cryptographic algorithms that should be implemented on the sensor: one for encryption and one for computing the MAC. This is the absolute minimum to ensure confidentiality and data authentication. Let us now briefly describe the most optimal choices for both types of algorithms in the context of 64-bit and 128-bit security.

- The standard encryption algorithm is AES [4], which can be used as a 128-, 190-, and 256-bit block cipher. Law et al. [12] have demonstrated that AES is the most appropriate block cipher algorithm when a high security level and energy efficiency are required in a wireless sensor network. An ultra-lightweight block cipher that is recently proposed in the context of RFID applications is PRESENT [1]. This block cipher has an 80-bit key and a block size of 64 bits, resulting in a security level of 64 bits.

- In [6], it is shown that the conventional MAC algorithms that are rec-

	64-bit security		128-bit security	
	PRESENT	TuLP	AES-128	TuLP-128
Functionality	Enc	MAC	Enc	MAC
Key size (bits)	80	80	128	160
Block size (bits)	64	64	128	128
RAM (bits)	1040	1048	1915	1056
ROM (bits)	1926	3302	12720	3718
Speed/8 byte (ms)	1.82	5.64	1.46	11.8

Table 2
Performance comparison of 64-bit and 128-bit security algorithms on MICAz node [6]

ommended for wireless sensor networks have some practical problems when applied in a WBSN. In their paper, they also introduce two new secure and practical MACs that are particularly designed for body sensor networks. The first MAC algorithm is TuLP and is based on the PRESENT block cipher, thus having 64-bit block size and 80-bit key size. The 128-bit variant is TuLP-128 and uses a 160-bit key.

Table 2 compares the parameters for these 64-bit and 128-bit algorithms, based on the results of [6]. Note that in order to compute the exact performance numbers, one can not simply add the numbers of the encryption algorithm PRESENT and the MAC algorithm TuLP, since the code of the latter contains large parts of the PRESENT algorithm. However, these figures give some estimation on the upperbound of the complexity. Comparing these upperbounds with the specification of the body sensor (see Table 1), we can conclude that both the 64-bit and 128-bit algorithm can be implemented on the MICAz node. On the other hand, the RAM capacity of the TI node is too small, even for the implementation with a security level of 64 bits.

Since complexity is shifted towards the patient's PDA, for scalability reasons, the computation and storage cost for the PDA is higher than the solution where the PDA just acts as a gateway. Concerning storage, the patient's PDA needs to store the ID and counter information for each sensor. Since the number of sensors in the network is very limited and the bitlength of both variables is rather small, this extra cost is negligible. On the other hand, the computation cost is increased with 1 symmetric key decryption, 1 public key encryption, and 1 signature for each transmission by a sensor. Let us illustrate

this with some numbers. We assume that 1024-bit RSA is used to compute the signatures, that there are 20 sensors in the WBSN that send information to the PDA once an hour. Based on the results of Potlapally et al. [21], the energy cost of computing a 1024-bit RSA signature is approximately 546 mJ. To carry out all the necessary cryptographic operations, the PDA will have an energy consumption of approximately 0.26 KJ on one day. To put this more into perspective, the typical energy content of a smartphone's battery is between 20 to 30 KJ.

However, we want to note that if you need to send frequently many data from the PDA to the server, it is best to use symmetric key cryptography to encrypt the data. One could use conventional key agreement protocols, such as the STS (Station-To-Station) protocol, to agree on a symmetric session key. In this case, the digital signature algorithm in protocol 2 is then replaced by a MAC algorithm. If the size of the data that has to be transmitted to the server is rather limited, then one could directly use a digital signature, as is discussed in this article.

5 RELATED WORK

One of the most crucial components to support the security architecture of a Wireless Sensor Network is its key management. There have been several proposals in literature [10, 19, 20].

As mentioned before, WBSNs differ from WSNs at several crucial points: number and range of the sensors, physical access to the sensors, and privacy constraints.

Several prototype implementations, specific for WBSNs, are already available in literature. However, very often studies on data security and privacy are missing in this area. An excellent recent survey on the general concept (applications, technologies, routing protocols, projects, etc.) of WBSNs can be found in [11].

Most articles on security and privacy of WBSNs published in the literature are either based on public key cryptography (e.g., [14, 26]), or on symmetric key cryptography in combination with physiological features (e.g., [2, 3, 15, 22, 23, 27, 7]) to commute the shared secret key. Using public key cryptography in WBSNs is not always a preferable solution due to the resource constraints on the body sensors. Deriving a shared secret key from physiological and/or behavioral characteristics is also not optimal since this approach requires statistical or machine learning techniques to recognize the medical data, which cannot be completely guaranteed.

Besides these two large categories of solutions, we are aware of only two other systems [24] where all cryptographic techniques for key distribution, key management and communication rely exclusively on symmetric key cryptography. However, the concept is different there, since it is required that all the medical data is first analyzed at the gateway (patient's PDA). In our setting, the analysis is left to the central server to reduce the computational overhead on the patient's PDA. Moreover, due to our particular construction of the shared secret key, we need less storage at the central server. This significantly improves the scalability. The system proposed in this article also includes an access control scheme based on time frames. The last difference is that in our scheme location privacy is directly enforced, while in [24] an add-on solution based on pseudonyms is proposed.

Note that in our work, we do do not focus on fine-grained distributed access control solutions, meaning that the patient's medical data is stored at several places, and that access is enforced using a role based access control model. The design challenges and a survey of such schemes can be found in [13].

6 CONCLUSION

We have presented an efficient and practical system to protect the communication flow of sensitive medical information in wireless body sensor networks. We proposed to construct a symmetric shared key, to secure the communication between a sensor and the central server, based on the central server's signature on the identity of the sensor. Such a construction drastically reduces the storage requirements at the central server. Our proposed solution also took into account the requirement of location privacy. This avoids that patients or doctors can be tracked by the devices they are carrying. Moreover, to enforce the minimal data disclosure principle, the doctor's access to the patient's medical data at the central server is restricted to a fixed number of time slots. Our proposed cryptographic protocols are secure, privacy-friendly, practical and efficient, and outperform the current state-of-the-art. It can directly be employed in a remote monitoring system based on wireless body sensor networks.

7 ACKNOWLEDGMENTS

This work was supported in part by the IAP Programme P6/26 BCRYPT of the Belgian State, by the Research Council K.U.Leuven: GOA TENSE,

and by the European Commission under contract number ICT-2007-216676 ECRYPT NoE phase II.

REFERENCES

[1] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe. (2007). PRESENT: An Ultra-Lightweight Block Cipher. In *Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems (CHES '07)*, Lecture Notes in Computer Science, LNCS 4727, pages 450–466. Springer-Verlag.

[2] F.M. Bui and D. Hatzinakos. (2008). Biometric Methods for Secure Communications in Body Sensor Networks: Resource-Efficient Key Management and Signal-Level Data Scrambling. *EURASIP Journal of Advanced Signal Process*, 8(2):1–16.

[3] S. Cherukuri, K.K. Venkatasubramanian, and S.K.S. Gupta. (2003). BioSec: A Biometric Based Approach for Securing Communication in Wireless Networks of Biosensors Implanted in the Human Body. In *Proceedings of the 2003 Workshop on Wireless Security and Privacy (WiSPr '03)*, pages 432–439. IEEE Computer Society.

[4] J. Daemen and V. Rijmen. (2002). *The Design of Rijndael – AES - The Advanced Encryption Standard*. Springer, 1st edition.

[5] S. Devadas, D. Clarke, B. Gassend, D. Lim, J. Lee, and M. van Dijk, (February 2007). Physical Unclonable Functions and Applications. MIT security lecture PUFs.

[6] Z. Gong, P. Hartel, S. Nikova, and B. Zhu. (2009). Towards Secure and Practical MACs for Body Sensor Networks. In *Proceedings of the 10th International Conference on Cryptology in India – Progress in Cryptology – INDOCRYPT '09*, Lecture Notes in Computer Science, LNCS 5922, pages 182–198. Springer-Verlag.

[7] M. Guennoun, M. Zandi, and K. El-Khatib. (2008). The Use of Biometrics to Secure Wireless Biosensor Networks. In *Proceedings of Information and Communication Technologies: From Theory to Applications*, pages 1–5. IEEE.

[8] M.A. Hanson, H.C. Powell, A.T. Barth, K. Ringgenberg, B.H. Calhoun, J.H. Aylor, and J. Lach. (2009). Body Area Sensor Networks: Challenges and Opportunities. *Computer*, 42(1):58–65.

[9] D. Henrici and P. Muller. (2004). Hash-Based Enhancement of Location Privacy for Radiofrequency Identification Devices Using Varying Identifiers. In *Proceedings of 2nd IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 149–153. IEEE Computer Society.

[10] C. Karlof, N. Sastry, and D. Wagner. (2004). TinySec: a Link Layer Security Architecture for Wireless Sensor Networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pages 162–175. ACM.

[11] B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester. (2010). A Survey on Wireless Body Area Networks. *Wireless Networks*, pages 1–18.

[12] Y.W. Law, J. Doumen, and P. Hartel. (2006). Survey and Benchmark of Block Ciphers for Wireless Sensor Networks. *ACM Transactions on Sensor Networks*, 2(1):65–93.

[13] M. Li and W. Lou. (2010). Data Security and Privacy in Wireless Body Area Networks. *IEEE Wireless Communications*, 17(1):51–58.

[14] K. Malasri and L. Wang. (2007). Addressing Security in Medical Sensor Networks. In *Proceedings of the 1st ACM SIGMOBILE International Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments*, pages 7–12. ACM.

[15] M. Mana, M. Feham, and B. Bensaber. (2009). SEKEBAN (Secure and Efficient Key Exchange for Wireless Body Area Network). *International Journal of Advanced Science and Technology*, 12:45–60.

[16] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. (1996). *Handbook of Applied Cryptography*. CRC Press, 5th edition.

[17] D. Molnar, A. Soppera, and D. Wagner. (2005). A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags. In *Proceedings of the 12th International Workshop on Selected Areas in Cryptography (SAC '05)*, Lecture Notes in Computer Science, LNCS 3897, pages 276–90. Springer-Verlag.

[18] P. Pecho, J. Nagy, P. Hanacek, and M. Drahansky. (2009). Secure Collection Tree Protocol for Tamper-Resistant Wireless Sensors. *Communications in Computer and Information Science*, 58:217–24.

[19] A. Perrig, J.A. Stankovic, and D. Wagner. (2004). Security in Wireless Sensor Networks. *Communications of the ACM*, 47(6):53–57.

[20] A. Perrig, R. Szewczyk, V. Wen, D.E. Culler, and J.D. Tygar. (2001). SPINS: Security Protocols for Sensor Networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking (MOBICOM '01)*, pages 189–199. ACM.

[21] N.R. Potlapally, S. Ravi, A. Raghunathan, and N.K. Jha. (2006). A Study of the Energy Consumption Characteristics of Cryptographic Algorithms and Security Protocols. *IEEE Transactions on Mobile Computing*, 5(2):128–143.

[22] S. Raazi and H. Lee. (2009). BARI: A Distributed Key Management Approach for Wireless Body Area Networks. In *Proceedings of 2009 International Conference on Computational Intelligence and Security*, pages 324–329. CPS.

[23] R. Sampangi, S. Dey, S. Urs, and S. Sampalli. (2012). IAMKeys: A Security Suite for Wireless Body Area Networks. *International journal of Network Security and its Applications*, 4(1):97–116.

[24] D. Singelée, B. Latré, B. Braem, M. Peeters, M. De Soete, P. De Cleyn, B. Preneel, I. Moerman, and C. Blondia. (2010). A Secure Low-Delay Protocol for Wireless Body Area Networks. *Ad-Hoc and Sensor Wireless Networks*, 9(1):53–72.

[25] G.E. Suh and S. Devadas. (2007). Physical Unclonable Functions for Device Authentication and Secret Key Generation. In *Proceedings of the 44th annual Design Automation Conference (DAC '07)*, pages 9–14. ACM.

[26] C.C. Tan, S. Zhong, H. Wang, and Q. Lin. (2008). Body Sensor Network Security: An Identity Based Cryptography Approach. In *Proceedings of the ACM Conference on Wireless Network Security (WISEC '08)*, pages 148–53. ACM.

[27] K. Venkatasubramanian, A. Banerjee, and S. Gupta. (2008). EKG-based Key Agreement in Body Sensor Networks. In *Proceedings of the IEEE INFOCOM Workshops 2008*, pages 1–6. IEEE Computer Society.

[28] L. Zhong, M. Sinclair, and R. Bittner. (2006). A Phone-Centered Body Sensor Network Platform: Cost, Energy Efficiency & User Interface. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN '06)*, pages 179–182. IEEE Computer Society.